

```
In [1]: #reference - https://colab.research.google.com/drive/140vFnAXggxB8vM4e8vSURUp1Tak
```

```
In [2]: # Install required packages.
!pip install -q torch-scatter -f https://pytorch-geometric.com/whl/torch-1.8.0+cu
!pip install -q torch-sparse -f https://pytorch-geometric.com/whl/torch-1.8.0+cu1
!pip install -q torch-geometric

# Helper function for visualization.
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.manifold import TSNE
```

```
In [3]: def visualize(h, color):
        z = TSNE(n_components=2).fit_transform(out.detach().cpu().numpy())

        plt.figure(figsize=(10,10))
        plt.xticks([])
        plt.yticks([])

        plt.scatter(z[:, 0], z[:, 1], s=70, c=color, cmap="Set2")
        plt.show()
```

```
In [4]: # Goal - node classification
# Input - a subset of nodes with ground truth labels
# Output - predict the labels of remaining nodes (transductive Learning)
# Dataset - Cora - a citation network where nodes represent documents.
#           Each node is described by a 1433-dimensional bag-of-words feature vec
#           Two documents are connected if there exists a citation link between t
#           The task is to infer the category of each document (7 in total)
```

```

In [5]: # Cora dataset is made available as part of PyG under the Planetoid name
from torch_geometric.datasets import Planetoid
from torch_geometric.transforms import NormalizeFeatures

dataset = Planetoid(root='data/Planetoid', name='Cora', transform=NormalizeFeatures)

# Data exploration
print()
print(f'Dataset: {dataset}:')
print('=====')
print(f'Number of graphs: {len(dataset)}')
print(f'Number of features: {dataset.num_features}')
print(f'Number of classes: {dataset.num_classes}')

data = dataset[0] # Get the first graph object.

print()
print(data)
print('=====')

# Graph statistics - total nodes = 2708, total edges = 10556
print(f'Number of nodes: {data.num_nodes}')
print(f'Number of edges: {data.num_edges}')
print(f'Average node degree: {data.num_edges / data.num_nodes:.2f}')
print(f'Number of training nodes: {data.train_mask.sum()}')
print(f'Training node label rate: {int(data.train_mask.sum()) / data.num_nodes:.2f}')
print(f'Contains isolated nodes: {data.contains_isolated_nodes()}')
print(f'Contains self-loops: {data.contains_self_loops()}')
print(f'Is undirected: {data.is_undirected()}')

```

Dataset: Cora():

=====

Number of graphs: 1

Number of features: 1433

Number of classes: 7

Data(edge_index=[2, 10556], test_mask=[2708], train_mask=[2708], val_mask=[2708], x=[2708, 1433], y=[2708])

=====

=====

Number of nodes: 2708

Number of edges: 10556

Average node degree: 3.90

Number of training nodes: 140

Training node label rate: 0.05

Contains isolated nodes: False

Contains self-loops: False

Is undirected: True

```

In [6]: # Train a multi layer perceptron, as just by checking the content of a document (
# it should be possible to categorize it

```

```

In [7]: import torch
        from torch.nn import Linear
        import torch.nn.functional as F

class MLP(torch.nn.Module):
    def __init__(self, hidden_channels):
        super(MLP, self).__init__()
        torch.manual_seed(12345)
        self.lin1 = Linear(dataset.num_features, hidden_channels)
        self.lin2 = Linear(hidden_channels, dataset.num_classes)

    def forward(self, x):
        x = self.lin1(x)
        x = x.relu()
        x = F.dropout(x, p=0.5, training=self.training)
        x = self.lin2(x)
        return x

model = MLP(hidden_channels=16)
print(model)

# model architecture
# layer 1 : reduce the 1433-dimensional feature vector to a low-dimensional embed
# layer 2 : Classifier that should map each resultant low-dimensional node embedd

MLP(
  (lin1): Linear(in_features=1433, out_features=16, bias=True)
  (lin2): Linear(in_features=16, out_features=7, bias=True)
)

```

```

In [8]: model = MLP(hidden_channels=16)
        criterion = torch.nn.CrossEntropyLoss() # Loss criterion.
        optimizer = torch.optim.Adam(model.parameters(), lr=0.01, weight_decay=5e-4) # c

def train():
    model.train()
    optimizer.zero_grad() # Clear gradients.
    out = model(data.x) # Perform a single forward pass.
    loss = criterion(out[data.train_mask], data.y[data.train_mask]) # Compute
    loss.backward() # Derive gradients.
    optimizer.step() # Update parameters based on gradients.
    return loss

def test():
    model.eval()
    out = model(data.x)
    pred = out.argmax(dim=1) # Use the class with highest probability.
    test_correct = pred[data.test_mask] == data.y[data.test_mask] # Check again
    test_acc = int(test_correct.sum()) / int(data.test_mask.sum()) # Derive re
    return test_acc

for epoch in range(1, 201):
    loss = train()
    print(f'Epoch: {epoch:03d}, Loss: {loss:.4f}')

```

```

Epoch: 001, Loss: 1.9615
Epoch: 002, Loss: 1.9557
Epoch: 003, Loss: 1.9505
Epoch: 004, Loss: 1.9423
Epoch: 005, Loss: 1.9327
Epoch: 006, Loss: 1.9279
Epoch: 007, Loss: 1.9144
Epoch: 008, Loss: 1.9087
Epoch: 009, Loss: 1.9023
Epoch: 010, Loss: 1.8893
Epoch: 011, Loss: 1.8776
Epoch: 012, Loss: 1.8594
Epoch: 013, Loss: 1.8457
Epoch: 014, Loss: 1.8365
Epoch: 015, Loss: 1.8280
Epoch: 016, Loss: 1.7965
Epoch: 017, Loss: 1.7984
Epoch: 018, Loss: 1.7832
Epoch: 019, Loss: 1.7495
Epoch: 020, Loss: 1.7444

```

```

In [9]: test_acc = test()
        print(f'Test Accuracy: {test_acc:.4f}')

```

```
Test Accuracy: 0.5900
```

```

In [10]: # Accuracy is pretty low :
         # possible causes - (i) small Labeled data leading to overfitting
         #                    (ii) Cited papers are very likely related to the category of

```

```
In [11]: from torch_geometric.nn import GCNConv

class GCN(torch.nn.Module):
    def __init__(self, hidden_channels):
        super(GCN, self).__init__()
        torch.manual_seed(12345)
        # below we convert to GNN by replacing Linear with GCNConv
        self.conv1 = GCNConv(dataset.num_features, hidden_channels)
        self.conv2 = GCNConv(hidden_channels, dataset.num_classes)

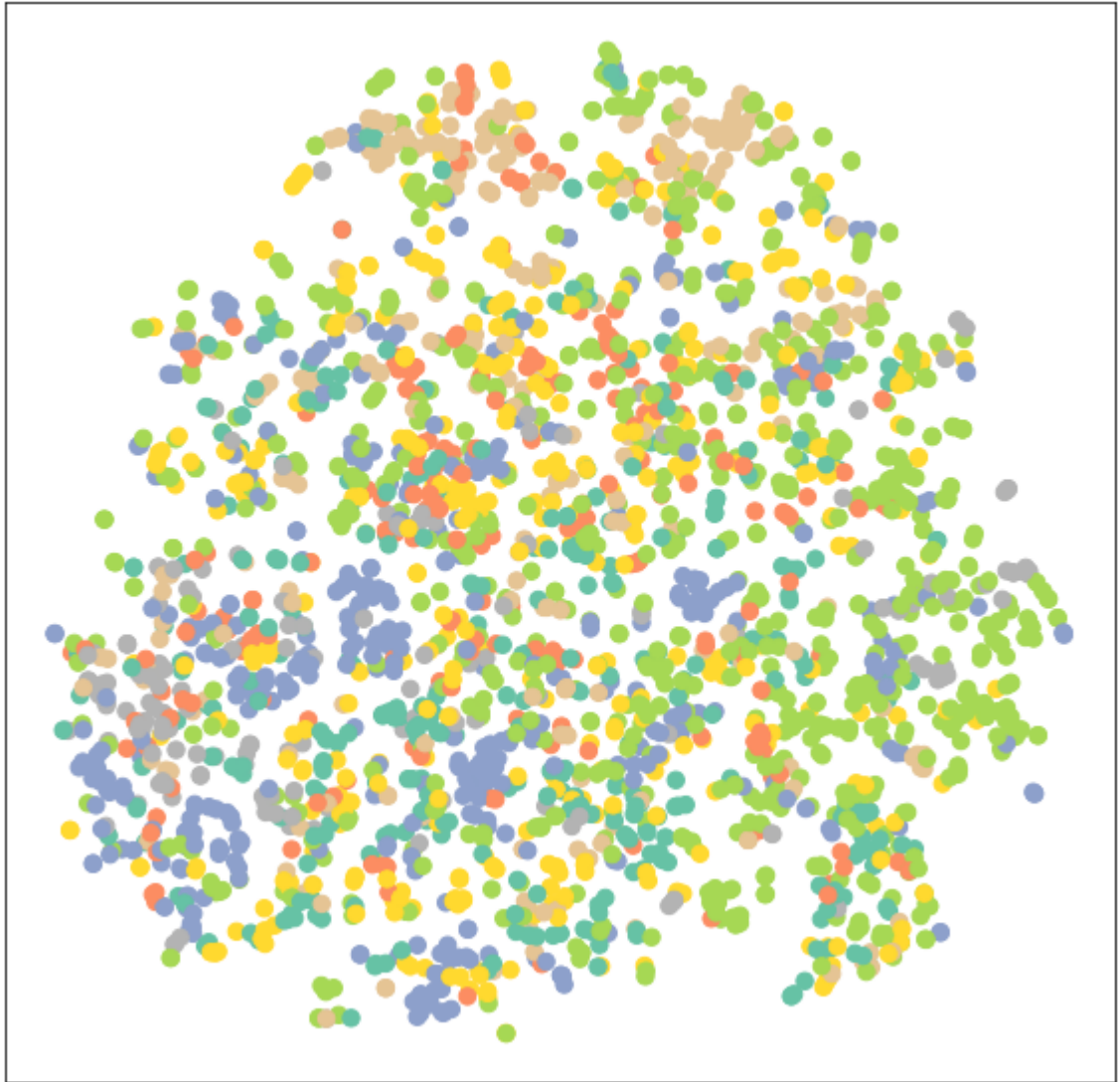
    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index)
        x = x.relu()
        x = F.dropout(x, p=0.5, training=self.training)
        x = self.conv2(x, edge_index)
        return x

model = GCN(hidden_channels=16)
print(model)
```

```
GCN(
  (conv1): GCNConv(1433, 16)
  (conv2): GCNConv(16, 7)
)
```

```
In [12]: # visualize the node embeddings of untrained GCN network - poor quality clustering
model = GCN(hidden_channels=16)
model.eval()

out = model(data.x, data.edge_index)
visualize(out, color=data.y)
```




```

In [13]: model = GCN(hidden_channels=16)
optimizer = torch.optim.Adam(model.parameters(), lr=0.01, weight_decay=5e-4) # optimizer
criterion = torch.nn.CrossEntropyLoss() # Loss criterion

def train():
    model.train()
    optimizer.zero_grad() # Clear gradients.
    out = model(data.x, data.edge_index) # Perform a single forward pass.
    loss = criterion(out[data.train_mask], data.y[data.train_mask]) # Compute loss
    loss.backward() # Derive gradients.
    optimizer.step() # Update parameters based on gradients.
    return loss

def test():
    model.eval()
    out = model(data.x, data.edge_index)
    pred = out.argmax(dim=1) # Use the class with highest probability.
    test_correct = pred[data.test_mask] == data.y[data.test_mask] # Check against ground truth
    test_acc = int(test_correct.sum()) / int(data.test_mask.sum()) # Derive accuracy
    return test_acc

for epoch in range(1, 201):
    loss = train()
    print(f'Epoch: {epoch:03d}, Loss: {loss:.4f}')

```

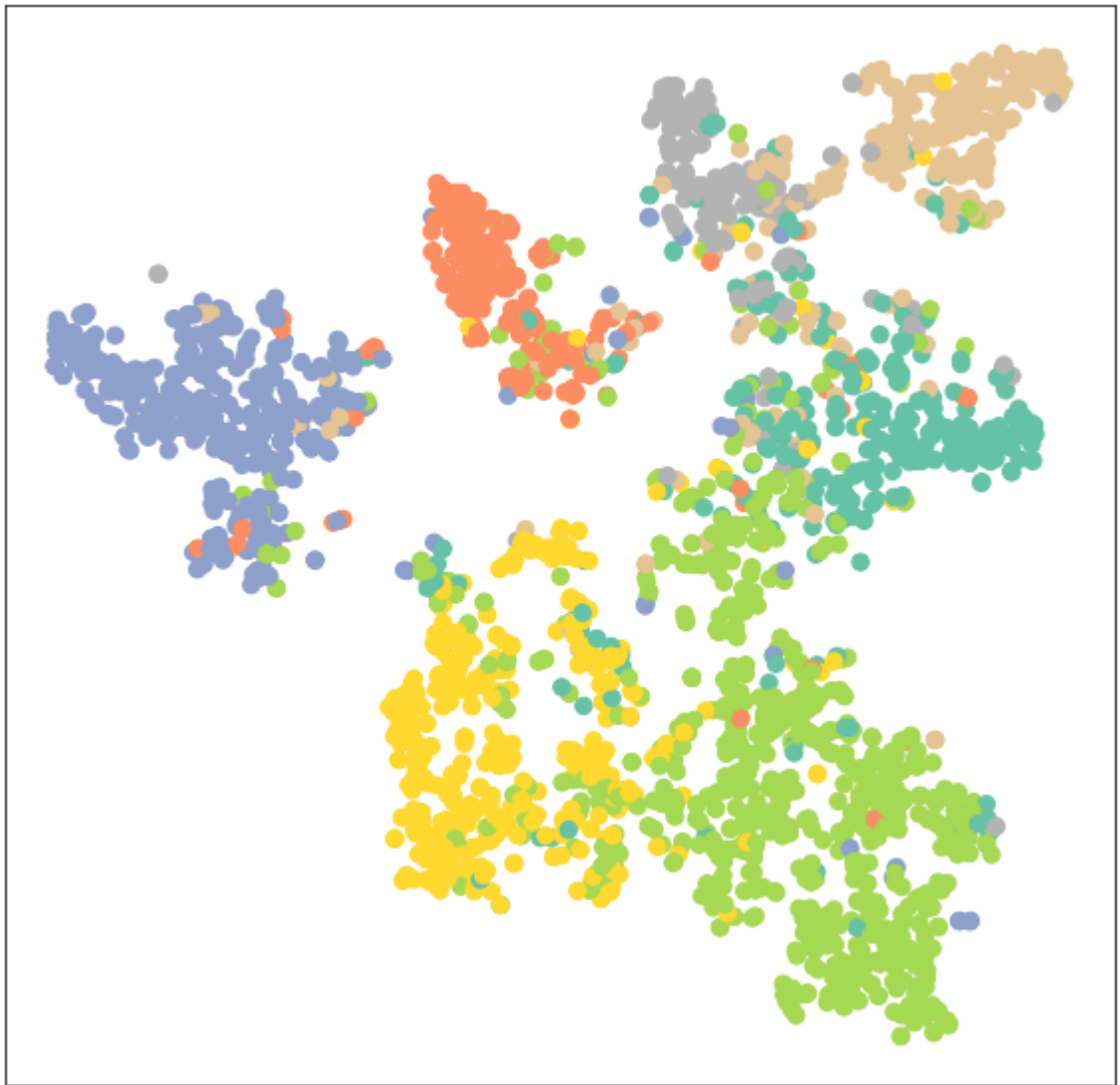
```

Epoch: 001, Loss: 1.9451
Epoch: 002, Loss: 1.9384
Epoch: 003, Loss: 1.9307
Epoch: 004, Loss: 1.9227
Epoch: 005, Loss: 1.9126
Epoch: 006, Loss: 1.9076
Epoch: 007, Loss: 1.8917
Epoch: 008, Loss: 1.8809
Epoch: 009, Loss: 1.8728
Epoch: 010, Loss: 1.8616
Epoch: 011, Loss: 1.8453
Epoch: 012, Loss: 1.8397
Epoch: 013, Loss: 1.8237
Epoch: 014, Loss: 1.8057
Epoch: 015, Loss: 1.7979
Epoch: 016, Loss: 1.7808
Epoch: 017, Loss: 1.7667
Epoch: 018, Loss: 1.7555
Epoch: 019, Loss: 1.7436
Epoch: 020, Loss: 1.7301
Epoch: 021, Loss: 1.7181
Epoch: 022, Loss: 1.7071
Epoch: 023, Loss: 1.6969
Epoch: 024, Loss: 1.6875
Epoch: 025, Loss: 1.6788
Epoch: 026, Loss: 1.6707
Epoch: 027, Loss: 1.6632
Epoch: 028, Loss: 1.6562
Epoch: 029, Loss: 1.6497
Epoch: 030, Loss: 1.6436
Epoch: 031, Loss: 1.6379
Epoch: 032, Loss: 1.6325
Epoch: 033, Loss: 1.6274
Epoch: 034, Loss: 1.6225
Epoch: 035, Loss: 1.6178
Epoch: 036, Loss: 1.6133
Epoch: 037, Loss: 1.6090
Epoch: 038, Loss: 1.6048
Epoch: 039, Loss: 1.6008
Epoch: 040, Loss: 1.5969
Epoch: 041, Loss: 1.5931
Epoch: 042, Loss: 1.5894
Epoch: 043, Loss: 1.5858
Epoch: 044, Loss: 1.5823
Epoch: 045, Loss: 1.5789
Epoch: 046, Loss: 1.5755
Epoch: 047, Loss: 1.5722
Epoch: 048, Loss: 1.5690
Epoch: 049, Loss: 1.5658
Epoch: 050, Loss: 1.5627
Epoch: 051, Loss: 1.5596
Epoch: 052, Loss: 1.5566
Epoch: 053, Loss: 1.5536
Epoch: 054, Loss: 1.5506
Epoch: 055, Loss: 1.5477
Epoch: 056, Loss: 1.5448
Epoch: 057, Loss: 1.5419
Epoch: 058, Loss: 1.5390
Epoch: 059, Loss: 1.5362
Epoch: 060, Loss: 1.5334
Epoch: 061, Loss: 1.5306
Epoch: 062, Loss: 1.5278
Epoch: 063, Loss: 1.5250
Epoch: 064, Loss: 1.5222
Epoch: 065, Loss: 1.5194
Epoch: 066, Loss: 1.5166
Epoch: 067, Loss: 1.5138
Epoch: 068, Loss: 1.5110
Epoch: 069, Loss: 1.5082
Epoch: 070, Loss: 1.5054
Epoch: 071, Loss: 1.5026
Epoch: 072, Loss: 1.5000
Epoch: 073, Loss: 1.4973
Epoch: 074, Loss: 1.4946
Epoch: 075, Loss: 1.4919
Epoch: 076, Loss: 1.4892
Epoch: 077, Loss: 1.4865
Epoch: 078, Loss: 1.4838
Epoch: 079, Loss: 1.4811
Epoch: 080, Loss: 1.4784
Epoch: 081, Loss: 1.4757
Epoch: 082, Loss: 1.4730
Epoch: 083, Loss: 1.4703
Epoch: 084, Loss: 1.4676
Epoch: 085, Loss: 1.4649
Epoch: 086, Loss: 1.4622
Epoch: 087, Loss: 1.4595
Epoch: 088, Loss: 1.4568
Epoch: 089, Loss: 1.4541
Epoch: 090, Loss: 1.4514
Epoch: 091, Loss: 1.4487
Epoch: 092, Loss: 1.4460
Epoch: 093, Loss: 1.4433
Epoch: 094, Loss: 1.4406
Epoch: 095, Loss: 1.4379
Epoch: 096, Loss: 1.4352
Epoch: 097, Loss: 1.4325
Epoch: 098, Loss: 1.4298
Epoch: 099, Loss: 1.4271
Epoch: 100, Loss: 1.4244
Epoch: 101, Loss: 1.4217
Epoch: 102, Loss: 1.4190
Epoch: 103, Loss: 1.4163
Epoch: 104, Loss: 1.4136
Epoch: 105, Loss: 1.4109
Epoch: 106, Loss: 1.4082
Epoch: 107, Loss: 1.4055
Epoch: 108, Loss: 1.4028
Epoch: 109, Loss: 1.4001
Epoch: 110, Loss: 1.3974
Epoch: 111, Loss: 1.3947
Epoch: 112, Loss: 1.3920
Epoch: 113, Loss: 1.3893
Epoch: 114, Loss: 1.3866
Epoch: 115, Loss: 1.3839
Epoch: 116, Loss: 1.3812
Epoch: 117, Loss: 1.3785
Epoch: 118, Loss: 1.3758
Epoch: 119, Loss: 1.3731
Epoch: 120, Loss: 1.3704
Epoch: 121, Loss: 1.3677
Epoch: 122, Loss: 1.3650
Epoch: 123, Loss: 1.3623
Epoch: 124, Loss: 1.3596
Epoch: 125, Loss: 1.3569
Epoch: 126, Loss: 1.3542
Epoch: 127, Loss: 1.3515
Epoch: 128, Loss: 1.3488
Epoch: 129, Loss: 1.3461
Epoch: 130, Loss: 1.3434
Epoch: 131, Loss: 1.3407
Epoch: 132, Loss: 1.3380
Epoch: 133, Loss: 1.3353
Epoch: 134, Loss: 1.3326
Epoch: 135, Loss: 1.3299
Epoch: 136, Loss: 1.3272
Epoch: 137, Loss: 1.3245
Epoch: 138, Loss: 1.3218
Epoch: 139, Loss: 1.3191
Epoch: 140, Loss: 1.3164
Epoch: 141, Loss: 1.3137
Epoch: 142, Loss: 1.3110
Epoch: 143, Loss: 1.3083
Epoch: 144, Loss: 1.3056
Epoch: 145, Loss: 1.3029
Epoch: 146, Loss: 1.3002
Epoch: 147, Loss: 1.2975
Epoch: 148, Loss: 1.2948
Epoch: 149, Loss: 1.2921
Epoch: 150, Loss: 1.2894
Epoch: 151, Loss: 1.2867
Epoch: 152, Loss: 1.2840
Epoch: 153, Loss: 1.2813
Epoch: 154, Loss: 1.2786
Epoch: 155, Loss: 1.2759
Epoch: 156, Loss: 1.2732
Epoch: 157, Loss: 1.2705
Epoch: 158, Loss: 1.2678
Epoch: 159, Loss: 1.2651
Epoch: 160, Loss: 1.2624
Epoch: 161, Loss: 1.2597
Epoch: 162, Loss: 1.2570
Epoch: 163, Loss: 1.2543
Epoch: 164, Loss: 1.2516
Epoch: 165, Loss: 1.2489
Epoch: 166, Loss: 1.2462
Epoch: 167, Loss: 1.2435
Epoch: 168, Loss: 1.2408
Epoch: 169, Loss: 1.2381
Epoch: 170, Loss: 1.2354
Epoch: 171, Loss: 1.2327
Epoch: 172, Loss: 1.2300
Epoch: 173, Loss: 1.2273
Epoch: 174, Loss: 1.2246
Epoch: 175, Loss: 1.2219
Epoch: 176, Loss: 1.2192
Epoch: 177, Loss: 1.2165
Epoch: 178, Loss: 1.2138
Epoch: 179, Loss: 1.2111
Epoch: 180, Loss: 1.2084
Epoch: 181, Loss: 1.2057
Epoch: 182, Loss: 1.2030
Epoch: 183, Loss: 1.2003
Epoch: 184, Loss: 1.1976
Epoch: 185, Loss: 1.1949
Epoch: 186, Loss: 1.1922
Epoch: 187, Loss: 1.1895
Epoch: 188, Loss: 1.1868
Epoch: 189, Loss: 1.1841
Epoch: 190, Loss: 1.1814
Epoch: 191, Loss: 1.1787
Epoch: 192, Loss: 1.1760
Epoch: 193, Loss: 1.1733
Epoch: 194, Loss: 1.1706
Epoch: 195, Loss: 1.1679
Epoch: 196, Loss: 1.1652
Epoch: 197, Loss: 1.1625
Epoch: 198, Loss: 1.1598
Epoch: 199, Loss: 1.1571
Epoch: 200, Loss: 1.1544
Epoch: 201, Loss: 1.1517
Epoch: 202, Loss: 1.1490
Epoch: 203, Loss: 1.1463
Epoch: 204, Loss: 1.1436
Epoch: 205, Loss: 1.1409
Epoch: 206, Loss: 1.1382
Epoch: 207, Loss: 1.1355
Epoch: 208, Loss: 1.1328
Epoch: 209, Loss: 1.1301
Epoch: 210, Loss: 1.1274
Epoch: 211, Loss: 1.1247
Epoch: 212, Loss: 1.1220
Epoch: 213, Loss: 1.1193
Epoch: 214, Loss: 1.1166
Epoch: 215, Loss: 1.1139
Epoch: 216, Loss: 1.1112
Epoch: 217, Loss: 1.1085
Epoch: 218, Loss: 1.1058
Epoch: 219, Loss: 1.1031
Epoch: 220, Loss: 1.1004
Epoch: 221, Loss: 1.0977
Epoch: 222, Loss: 1.0950
Epoch: 223, Loss: 1.0923
Epoch: 224, Loss: 1.0896
Epoch: 225, Loss: 1.0869
Epoch: 226, Loss: 1.0842
Epoch: 227, Loss: 1.0815
Epoch: 228, Loss: 1.0788
Epoch: 229, Loss: 1.0761
Epoch: 230, Loss: 1.0734
Epoch: 231, Loss: 1.0707
Epoch: 232, Loss: 1.0680
Epoch: 233, Loss: 1.0653
Epoch: 234, Loss: 1.0626
Epoch: 235, Loss: 1.0599
Epoch: 236, Loss: 1.0572
Epoch: 237, Loss: 1.0545
Epoch: 238, Loss: 1.0518
Epoch: 239, Loss: 1.0491
Epoch: 240, Loss: 1.0464
Epoch: 241, Loss: 1.0437
Epoch: 242, Loss: 1.0410
Epoch: 243, Loss: 1.0383
Epoch: 244, Loss: 1.0356
Epoch: 245, Loss: 1.0329
Epoch: 246, Loss: 1.0302
Epoch: 247, Loss: 1.0275
Epoch: 248, Loss: 1.0248
Epoch: 249, Loss: 1.0221
Epoch: 250, Loss: 1.0194
Epoch: 251, Loss: 1.0167
Epoch: 252, Loss: 1.0140
Epoch: 253, Loss: 1.0113
Epoch: 254, Loss: 1.0086
Epoch: 255, Loss: 1.0059
Epoch: 256, Loss: 1.0032
Epoch: 257, Loss: 1.0005
Epoch: 258, Loss: 0.9978
Epoch: 259, Loss: 0.9951
Epoch: 260, Loss: 0.9924
Epoch: 261, Loss: 0.9897
Epoch: 262, Loss: 0.9870
Epoch: 263, Loss: 0.9843
Epoch: 264, Loss: 0.9816
Epoch: 265, Loss: 0.9789
Epoch: 266, Loss: 0.9762
Epoch: 267, Loss: 0.9735
Epoch: 268, Loss: 0.9708
Epoch: 269, Loss: 0.9681
Epoch: 270, Loss: 0.9654
Epoch: 271, Loss: 0.9627
Epoch: 272, Loss: 0.9600
Epoch: 273, Loss: 0.9573
Epoch: 274, Loss: 0.9546
Epoch: 275, Loss: 0.9519
Epoch: 276, Loss: 0.9492
Epoch: 277, Loss: 0.9465
Epoch: 278, Loss: 0.9438
Epoch: 279, Loss: 0.9411
Epoch: 280, Loss: 0.9384
Epoch: 281, Loss: 0.9357
Epoch: 282, Loss: 0.9330
Epoch: 283, Loss: 0.9303
Epoch: 284, Loss: 0.9276
Epoch: 285, Loss: 0.9249
Epoch: 286, Loss: 0.9222
Epoch: 287, Loss: 0.9195
Epoch: 288, Loss: 0.9168
Epoch: 289, Loss: 0.9141
Epoch: 290, Loss: 0.9114
Epoch: 291, Loss: 0.9087
Epoch: 292, Loss: 0.9060
Epoch: 293, Loss: 0.9033
Epoch: 294, Loss: 0.9006
Epoch: 295, Loss: 0.8979
Epoch: 296, Loss: 0.8952
Epoch: 297, Loss: 0.8925
Epoch: 298, Loss: 0.8898
Epoch: 299, Loss: 0.8871
Epoch: 300, Loss: 0.8844
Epoch: 301, Loss: 0.8817
Epoch: 302, Loss: 0.8790
Epoch: 303, Loss: 0.8763
Epoch: 304, Loss: 0.8736
Epoch: 305, Loss: 0.8709
Epoch: 306, Loss: 0.8682
Epoch: 307, Loss: 0.8655
Epoch: 308, Loss: 0.8628
Epoch: 309, Loss: 0.8601
Epoch: 310, Loss: 0.8574
Epoch: 311, Loss: 0.8547
Epoch: 312, Loss: 0.8520
Epoch: 313, Loss: 0.8493
Epoch: 314, Loss: 0.8466
Epoch: 315, Loss: 0.8439
Epoch: 316, Loss: 0.8412
Epoch: 317, Loss: 0.8385
Epoch: 318, Loss: 0.8358
Epoch: 319, Loss: 0.8331
Epoch: 320, Loss: 0.8304
Epoch: 321, Loss: 0.8277
Epoch: 322, Loss: 0.8250
Epoch: 323, Loss: 0.8223
Epoch: 324, Loss: 0.8196
Epoch: 325, Loss: 0.8169
Epoch: 326, Loss: 0.8142
Epoch: 327, Loss: 0.8115
Epoch: 328, Loss: 0.8088
Epoch: 329, Loss: 0.8061
Epoch: 330, Loss: 0.8034
Epoch: 331, Loss: 0.8007
Epoch: 332, Loss: 0.7980
Epoch: 333, Loss: 0.7953
Epoch: 334, Loss: 0.7926
Epoch: 335, Loss: 0.7899
Epoch: 336, Loss: 0.7872
Epoch: 337, Loss: 0.7845
Epoch: 338, Loss: 0.7818
Epoch: 339, Loss: 0.7791
Epoch: 340, Loss: 0.7764
Epoch: 341, Loss: 0.7737
Epoch: 342, Loss: 0.7710
Epoch: 343, Loss: 0.7683
Epoch: 344, Loss: 0.7656
Epoch: 345, Loss: 0.7629
Epoch: 346, Loss: 0.7602
Epoch: 347, Loss: 0.7575
Epoch: 348, Loss: 0.7548
Epoch: 349, Loss: 0.7521
Epoch: 350, Loss: 0.7494
Epoch: 351, Loss: 0.7467
Epoch: 352, Loss: 0.7440
Epoch: 353, Loss: 0.7413
Epoch: 354, Loss: 0.7386
Epoch: 355, Loss: 0.7359
Epoch: 356, Loss: 0.7332
Epoch: 357, Loss: 0.7305
Epoch: 358, Loss: 0.7278
Epoch: 359, Loss: 0.7251
Epoch: 360, Loss: 0.7224
Epoch: 361, Loss: 0.7197
Epoch: 362, Loss: 0.7170
Epoch: 363, Loss: 0.7143
Epoch: 364, Loss: 0.7116
Epoch: 365, Loss: 0.7089
Epoch: 366, Loss: 0.7062
Epoch: 367, Loss: 0.7035
Epoch: 368, Loss: 0.7008
Epoch: 369, Loss: 0.6981
Epoch: 370, Loss: 0.6954
Epoch: 371, Loss: 0.6927
Epoch: 372, Loss: 0.6900
Epoch: 373, Loss: 0.6873
Epoch: 374, Loss: 0.6846
Epoch: 375, Loss: 0.6819
Epoch: 376, Loss: 0.6792
Epoch: 377, Loss: 0.6765
Epoch: 378, Loss: 0.6738
Epoch: 379, Loss: 0.6711
Epoch: 380, Loss: 0.6684
Epoch: 381, Loss: 0.6657
Epoch: 382, Loss: 0.6630
Epoch: 383, Loss: 0.6603
Epoch: 384, Loss: 0.6576
Epoch: 385, Loss: 0.6549
Epoch: 386, Loss: 0.6522
Epoch: 387, Loss: 0.6495
Epoch: 388, Loss: 0.6468
Epoch: 389, Loss: 0.6441
Epoch: 390, Loss: 0.6414
Epoch: 391, Loss: 0.6387
Epoch: 392, Loss: 0.6360
Epoch: 393, Loss: 0.6333
Epoch: 394, Loss: 0.6306
Epoch: 395, Loss: 0.6279
Epoch: 396, Loss: 0.6252
Epoch: 397, Loss: 0.6225
Epoch: 398, Loss: 0.6198
Epoch: 399, Loss: 0.6171
Epoch: 400, Loss: 0.6144
Epoch: 401, Loss: 0.6117
Epoch: 402, Loss: 0.6090
Epoch: 403, Loss: 0.6063
Epoch: 404, Loss: 0.6036
Epoch: 405, Loss: 0.6009
Epoch: 406, Loss: 0.5982
Epoch: 407, Loss: 0.5955
Epoch: 408, Loss: 0.5928
Epoch: 409, Loss: 0.5901
Epoch: 410, Loss: 0.5874
Epoch: 411, Loss: 0.5847
Epoch: 412, Loss: 0.5820
Epoch: 413, Loss: 0.5793
Epoch: 414, Loss: 0.5766
Epoch: 415, Loss: 0.5739
Epoch: 416, Loss: 0.5712
Epoch: 417, Loss: 0.5685
Epoch: 418, Loss: 0.5658
Epoch: 419, Loss: 0.5631
Epoch: 420, Loss: 0.5604
Epoch: 421, Loss: 0.5577
Epoch: 422, Loss: 0.5550
Epoch: 423, Loss: 0.5523
Epoch: 424, Loss: 0.5496
Epoch: 425, Loss: 0.5469
Epoch: 426, Loss: 0.5442
Epoch: 427, Loss: 0.5415
Epoch: 428, Loss: 0.5388
Epoch: 429, Loss: 0.5361
Epoch: 430, Loss: 0.5334
Epoch: 431, Loss: 0.5307
Epoch: 432, Loss: 0.5280
Epoch: 433, Loss: 0.5253
Epoch: 434, Loss: 0.5226
Epoch: 435, Loss: 0.5199
Epoch: 436, Loss: 0.5172
Epoch: 437, Loss: 0.5145
Epoch: 438, Loss: 0.5118
Epoch: 439, Loss: 0.5091
Epoch: 440, Loss: 0.5064
Epoch: 441, Loss: 0.5037
Epoch: 442, Loss: 0.5010
Epoch: 443, Loss: 0.4983
Epoch: 444, Loss: 0.4956
Epoch: 445, Loss: 0.4929
Epoch: 446, Loss: 0.4902
Epoch: 447, Loss: 0.4875
Epoch: 448, Loss: 0.4848
Epoch: 449, Loss: 0.4821
Epoch: 450, Loss: 0.4794
Epoch: 451, Loss: 0.4767
Epoch: 452, Loss: 0.4740
Epoch: 453, Loss: 0.4713
Epoch: 454, Loss: 0.4686
Epoch: 455, Loss: 0.4659
Epoch: 456, Loss: 0.4632
Epoch: 457, Loss: 0.4605
Epoch: 458, Loss: 0.4578
Epoch: 459, Loss: 0.4551
Epoch: 460, Loss: 0.4524
Epoch: 461, Loss: 0.4497
Epoch: 462, Loss: 0.4470
Epoch: 463, Loss: 0.4443
Epoch: 464, Loss: 0.4416
Epoch: 465, Loss: 0.4389
Epoch: 466, Loss: 0.4362
Epoch: 467, Loss: 0.4335
Epoch: 468, Loss: 0.4308
Epoch: 469, Loss: 0.4281
Epoch: 470, Loss: 0.4254
Epoch: 471, Loss: 0.4227
Epoch: 472, Loss: 0.4200
Epoch: 473, Loss: 0.4173
Epoch: 474, Loss: 0.4146
Epoch: 475, Loss: 0.4119
Epoch: 476, Loss: 0.4092
Epoch: 477, Loss: 0.4065
Epoch: 478, Loss: 0.4038
Epoch: 479, Loss: 0.4011
Epoch: 480, Loss: 0.3984
Epoch: 481, Loss: 0.3957
Epoch: 482, Loss: 0.3930
Epoch: 483, Loss: 0.3903
Epoch: 484, Loss: 0.3876
Epoch: 485, Loss: 0.3849
Epoch: 486, Loss: 0.3822
Epoch: 487, Loss: 0.3795
Epoch: 488, Loss: 0.3768
Epoch: 489, Loss: 0.3741
Epoch: 490, Loss: 0.3714
Epoch: 491, Loss: 0.3687
Epoch: 492, Loss: 0.3660
Epoch: 493, Loss: 0.3633
Epoch: 494, Loss: 0.3606
Epoch: 495, Loss: 0.3579
Epoch: 496, Loss: 0.3552
Epoch: 497, Loss: 0.3525
Epoch: 498, Loss: 0.3498
Epoch: 499, Loss: 0.3471
Epoch: 500, Loss: 0.3444
Epoch: 501, Loss: 0.3417
Epoch: 502, Loss: 0.3390
Epoch: 503, Loss: 0.3363
Epoch: 504, Loss: 0.3336
Epoch: 505, Loss: 0.3309
Epoch: 506, Loss: 0.3282
Epoch: 507, Loss: 0.3255
Epoch: 508, Loss: 0.3228
Epoch: 509, Loss: 0.3201
Epoch: 510, Loss: 0.3174
Epoch: 511, Loss: 0.3147
Epoch: 512, Loss: 0.3120
Epoch: 513, Loss: 0.3093
Epoch: 514, Loss: 0.3066
Epoch: 515, Loss: 0.3039
Epoch: 516, Loss: 0.3012
Epoch: 517, Loss: 0.2985
Epoch: 518, Loss: 0.2958
Epoch: 519, Loss: 0.2931
Epoch: 520, Loss: 0.2904
Epoch: 521, Loss: 0.2877
Epoch: 522, Loss: 0.2850
Epoch: 523, Loss: 0.2823
Epoch: 524, Loss: 0.2796
Epoch: 525, Loss: 0.2769
Epoch: 526, Loss: 0.2742
Epoch: 527, Loss: 0.2715
Epoch: 528, Loss: 0.2688
Epoch: 529, Loss: 0.2661
Epoch: 530, Loss: 0.2634
Epoch: 531, Loss: 0.2607
Epoch: 532, Loss: 0.2580
Epoch: 533, Loss: 0.2553
Epoch: 534, Loss: 0.2526
Epoch: 535, Loss: 0.2499
Epoch: 536, Loss: 0.2472
Epoch: 537, Loss: 0.2445
Epoch: 538, Loss: 0.2418
Epoch: 539, Loss: 0.2391
Epoch: 540, Loss: 0.2364
Epoch: 541, Loss: 0.2337
Epoch: 542, Loss: 0.2310
Epoch: 543, Loss: 0.2283
Epoch: 544, Loss: 0.2256
Epoch: 545, Loss: 0.2229
Epoch: 546, Loss: 0.2202
Epoch: 547, Loss: 0.2175
Epoch: 548, Loss: 0.2148
Epoch: 549, Loss: 0.2121
Epoch: 550, Loss: 0.2094
Epoch: 551, Loss: 0.2067
Epoch: 552, Loss: 0.2040
Epoch: 553, Loss: 0.2013
Epoch: 554, Loss: 0.1986
Epoch: 555, Loss: 0.1959
Epoch: 556, Loss: 0.1932
Epoch: 557, Loss: 0.1905
Epoch: 558, Loss: 0.1878
Epoch: 559, Loss: 0.1851
Epoch: 560, Loss: 0.1824
Epoch: 561, Loss: 0.1797
Epoch: 562, Loss: 0.1770
Epoch: 563, Loss: 0.1743
Epoch: 564, Loss: 0.1716
Epoch: 565, Loss: 0.1689
Epoch: 566, Loss: 0.1662
Epoch: 567, Loss: 0.1635
Epoch: 568, Loss: 0.1608
Epoch: 569, Loss: 0.1581
Epoch: 570, Loss: 0.1554
Epoch: 571, Loss: 0.1527
Epoch: 572, Loss: 0.1500
Epoch: 573, Loss: 0.1473
Epoch: 574, Loss: 0.1446
Epoch: 575, Loss: 0.1419
Epoch: 576, Loss: 0.1392
Epoch: 577, Loss: 0.1365
Epoch: 578, Loss: 0.1338
Epoch: 579, Loss: 0.1311
Epoch: 580, Loss: 0.1284
Epoch: 581, Loss: 0.1257
Epoch: 582, Loss: 0.1230
Epoch: 583, Loss: 0.1203
Epoch: 584, Loss: 0.1176
Epoch: 585, Loss: 0.1149
Epoch: 586, Loss: 0.1122
Epoch: 587, Loss: 0.1095
Epoch: 588, Loss: 0.1068
Epoch: 589, Loss: 0.1041
Epoch: 590, Loss: 0.1014
Epoch: 591, Loss: 0.0987
Epoch: 592, Loss: 0.0960
Epoch: 593, Loss: 0.0933
Epoch: 594, Loss: 0.0906
Epoch: 595, Loss: 0.0879
Epoch: 596, Loss: 0.0852
Epoch: 597, Loss: 0.0825
Epoch: 598, Loss: 0.0798
Epoch: 599, Loss: 0.0771
Epoch: 600, Loss: 0.0744
Epoch: 601, Loss: 0.0717
Epoch: 602, Loss: 0.0690
Epoch: 603, Loss: 0.0663
Epoch: 604, Loss: 0.0636
Epoch: 605, Loss: 0.0609
Epoch: 606, Loss: 0.0582
Epoch: 607, Loss: 0.0555
Epoch: 608, Loss: 0.0528
Epoch: 609, Loss: 0.0501
Epoch: 610, Loss: 0.0474
Epoch: 611, Loss: 0.0447
Epoch: 612, Loss: 0.0420
Epoch: 613, Loss: 0.0393
Epoch: 614, Loss: 0.0366
Epoch: 615, Loss: 0.0339
Epoch: 616, Loss: 0.0312
Epoch: 617, Loss: 0.0285
Epoch: 618, Loss: 0.0258
Epoch: 619, Loss: 0.0231
Epoch: 620, Loss: 0.0204
Epoch: 621, Loss: 0.0177
Epoch: 622, Loss: 0.0150
Epoch: 623, Loss: 0.0123
Epoch: 624, Loss: 0.0096
Epoch: 625, Loss: 0.0069
Epoch: 626, Loss: 0.0042
Epoch: 627, Loss: 0.0015
Epoch: 628, Loss: -0.0012
Epoch: 629, Loss: -0.0039
Epoch: 630, Loss: -0.0066
Epoch: 631, Loss: -0.0093
Epoch: 632, Loss: -0.0120
Epoch: 633, Loss: -0.0147
Epoch: 634, Loss: -0.0174
Epoch: 635, Loss: -0.0201
Epoch: 636, Loss: -0.0228
Epoch: 637, Loss: -0.0255
Epoch: 638, Loss: -0.0282
Epoch: 639, Loss: -0.0309
Epoch: 640, Loss: -0.0336
Epoch: 641, Loss: -0.0363
Epoch: 642, Loss: -0.0390
Epoch: 643, Loss: -0.0417
Epoch: 644, Loss: -0.0444
Epoch: 645, Loss: -0.0471
Epoch: 646, Loss: -0.0498
Epoch: 647, Loss: -0.0525
Epoch: 648, Loss: -0.0552
Epoch: 649, Loss: -0.0579
Epoch: 650, Loss: -0.0606
Epoch: 651, Loss: -0.0633
Epoch: 652, Loss: -0.0660
Epoch: 653, Loss: -0.0687
Epoch: 654, Loss: -0.0714
Epoch: 655, Loss: -0.0741
Epoch: 656, Loss: -0.0768
Epoch: 657, Loss: -0.0795
Epoch: 658, Loss: -0.0822
Epoch: 659, Loss: -0.0849
Epoch: 660, Loss: -0.0876
Epoch: 661, Loss: -0.0903
Epoch: 662, Loss: -0.0930
Epoch: 663, Loss: -0.0957
Epoch: 664, Loss: -0.0984
Epoch: 665, Loss: -0.1011
Epoch: 666, Loss: -0.1038
Epoch: 667, Loss: -0.1065
Epoch: 668, Loss: -0.1092
Epoch: 669, Loss: -0.1119
Epoch: 670, Loss: -0.1146
Epoch: 671, Loss: -0.1173
Epoch: 672, Loss: -0.1200
Epoch: 673, Loss: -0.1227
Epoch: 674, Loss: -0.1254
Epoch: 675, Loss: -0.1281
Epoch: 676, Loss: -0.1308
Epoch: 677, Loss: -0.1335
Epoch: 678, Loss: -0.1362
Epoch: 679, Loss: -0.1389
Epoch: 680, Loss: -0.1416
Epoch: 681, Loss: -0.1443
Epoch: 682, Loss: -0.1470
Epoch: 683, Loss: -0.1497
Epoch: 684, Loss: -0.1524
Epoch: 685, Loss: -0.1551
Epoch: 686, Loss: -0.1578
Epoch: 687, Loss: -0.1605
Epoch: 688, Loss: -0.1632
Epoch: 689, Loss: -0.1659
Epoch: 690, Loss: -0.1686
Epoch: 691, Loss: -0.1713
Epoch: 692, Loss: -0.1740
Epoch: 693, Loss: -0.1767
Epoch: 694, Loss: -0.1794
Epoch: 695, Loss: -0.1821
Epoch: 696, Loss: -0.1848
Epoch: 697, Loss: -0.1875
Epoch: 698, Loss: -0.1902
Epoch: 699, Loss: -0.1929
Epoch: 700, Loss: -0.1956
Epoch: 701, Loss: -0.1983
Epoch: 702, Loss: -0.2010
Epoch: 703, Loss: -0.2037
Epoch: 704, Loss: -0.2064
Epoch: 
```



```
In [15]: # visualize the node embeddings of trained GCN network - improved quality cluster
model.eval()

out = model(data.x, data.edge_index)
visualize(out, color=data.y)
```



In []: